Chips-dips

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Aibar has come up with a new board game and invited you to try it. The game set consists of n game chips, each chip having a unique rank from 1 to n. Aibar has already set up chips in a line in certain initial ordering and told you the final ordering.

Your task is to move the chips one by one in order to achieve the final ordering. When moving a chip to a new position, all the chips between the old and the new position of the moved chip are considered **activated** for the duration of the current move.

The game has only one rule: to move a chip of rank x to a new position, the number of **activated** chips of rank higher than x must be equal to the number of **activated** chips of rank lower than x.

For example, suppose the chips are ordered as $[8, 4, 5, 6, 3, 2, 7, 1]$. You may move the chip at the 2-nd position 4 positions to the right. Then the chips $\{5, 6, 3, 2\}$ will be activated, where 5 and 6 are of higher rank than 4, while 3 and 2 are of lower. There is the same number of both so the move is legal and the new ordering will be $[8, 5, 6, 3, 2, 4, 7, 1]$.

Find a sequence of moves leading to the final ordering.

Input

The first input line contains two numbers n and t ($1 \le n \le 200, 1 \le t \le 2$) — the number of gaming chips and the required output type.

The next two lines each contain a permutation of numbers from 1 to n — the initial and final orderings respectively.

Output

Print `YES`, if it is possible to win, that is to arrange the chips as in final ordering, or `NO` otherwise.

If $t = 2$ and it is possible to win, print m ($0 \le m \le 10^6$) — the number of required moves — in the next line.

Each of the next m lines must contain two numbers p and k ($1 \le p \le n$, $1 \le k \le n - i$ or $1 \le -k \le i - 1$) describing the according move: the chip at the position p is moved k positions to the right. If $k < 0$, then chip is moved $|k|$ positions to the left.

It is possible to show that whenever a winning strategy exists under the given constraints, there is a sequence of moves leading to the final ordering of length at most 10^6. You may print any such sequence.

Scoring

This problem is made up of 8 subtasks, that meet the following constraints:

1. $n \le 8$. Worth 7 points.
2. $n \le 11$. Worth 12 points.
3. It is guaranteed, that there exists a number k such that $a_i = i$ ($1 \le i \le k-1$), $a_j = j$ ($k+2 \le i \le n$) and $a_k = k + 1$, $a_{k+1} = k$, where a_i denotes the rank of i-th chip in the initial ordering. Worth 9 points.
4. $t = 1$. Worth 12 points.
5. $n \le 25$. Worth 11 points.

6. $n \leq 50$. Worth 18 points.

7. $n \leq 100$. Worth 17 points.

8. Original problem constraints. Worth 14 points.

Examples

standard input	standard output
3 1 1 2 3 3 2 1	NO
4 2 3 2 4 1 4 2 1 3	YES 3 2 2 1 2 4 -2
8 2 5 7 6 8 1 3 2 4 8 4 3 5 6 7 1 2	YES 5 2 2 6 -2 2 2 1 2 8 -6